

**Министерство образования и науки Российской Федерации  
Федеральное государственное учреждение высшего образования  
Московский государственный университет технологий и управления  
(первый казачий университет)  
Университетский колледж информационных технологий**

**Специальность 09.02.03 Программирование в компьютерных системах**

**ПМ1 Разработка программных модулей программного обеспечения компьютерных сетей**

**МДК1.2 Прикладное программирование Сборник заданий для практических работ (1 часть)**

## Содержание

1	Аннотация	3
2	Введение	4
3	Практическая работа №1. Простые программы с использованием событийно-ориентированного программирования (2 часа)	5
4	Практическая работа №2. Программирование графики и таймера (6 часов)	8
5	Практическая работа №3. Сложные элементы управления (4 часа)	17
6	Практическая работа №4. Динамическое создание элементов управления (4 часа)	22

# 1 Аннотация

Сборник заданий содержит задания для практических работ по МДК «Прикладное программирование» для первой половины курса (16 часов практических работ).

Представленные работы направлены на получение практического опыта

- разработки алгоритма поставленной задачи;
- разработки кода программного продукта;
- использования инструментальных средств на этапе отладки программного продукта;

умения

- осуществлять разработку кода программного модуля на современных языках программирования;
- создавать программу по разработанному алгоритму как отдельный модуль;
- выполнять отладку и тестирование программы на уровне модуля;

закрепление знаний

- основных этапов разработки программного обеспечения;
- основных принципов технологии структурного и объектно-ориентированного программирования;
- основных принципов отладки и тестирования программных продуктов.

Представленные задания направлены на формирование следующих профессиональных компетенций:

ПК1.2 осуществлять разработку кода программного продукта на основе готовых спецификаций на уровне модуля;

ПК1.3 выполнять отладку программных модулей с использованием специализированных программных средств;

ПК1.4 выполнять тестирование программных модулей;

ПК1.5 осуществлять оптимизацию программного кода модуля;

и общих компетенций:

ОК1 понимать сущность и социальную значимость своей будущей профессии, проявлять к ней устойчивый интерес;

ОК2 организовывать собственную деятельность, выбирать типовые методы и способы выполнения профессиональных задач, оценивать их эффективность и качество;

ОК3 принимать решения в стандартных и нестандартных ситуациях и нести за них ответственность;

ОК4 осуществлять поиск и использование информации, необходимой для эффективного выполнения профессиональных задач, профессионального и личностного развития;

ОК5 использование информационно-коммуникационных технологий в профессиональной деятельности;

ОК7 брать на себя ответственность за работу членов команды (подчиненных), за результат выполнения заданий.

## 2 Введение

Все представленные в задачнике задания необходимо выполнять в среде, предоставляющей возможность визуального проектирования пользовательского оконного интерфейса. По умолчанию, предполагается, что студенты используют интегрированную среду разработки (далее, IDE) Lazarus. Допустимо также использовать Visual Studio (.NET); QT Creator; а также Delphi (в случае соблюдения условий лицензионных соглашений).

Задания оцениваются по двум основным параметрам: уровню сложности задания, своевременности его выполнения, а также по показателям качества.

Следующие требования являются обязательными, при этом **жирным шрифтом** выделены те требования, которые специфичны для используемых IDE.

### 1. Общие требования к исходному коду:

- (a) все комментарии, описанные ниже, где это применимо должны удовлетворять формату ROBODOC (doxygen для C);
- (b) в начале каждого файла должен быть комментарий, содержащий сведения об авторе работы, номере задания и варианта, краткой формулировки задания (или его части);
- (c) к неочевидным действиям должны быть указаны комментарии (проверяется методом экспертной оценки лицом, осуществляющим проверку);
- (d) для каждой подпрограммы (**включая методы, созданные без использования мастера создания методов**) должен быть указан комментарий, содержащий полное описание ее работы, описание всех аргументов и результатов (этот комментарий должен быть достаточен для возможности использовать подпрограмму в других программах (без изучения собственно текста подпрограммы));
- (e) для каждой глобальной переменной **поля, свойства и метода класса** должно быть указано ее назначение;
- (f) для всех переменных, кроме переменных цикла, должны быть использованы «говорящие» (англоязычные!) названий;
- (g) **все поля комбинированного типа, свойства, поля и методы класса**
- (h) для всех подпрограмм (**включая методы**) должны быть использованы говорящие (англоязычные!) названия;
- (i) на одной строке должен быть написан один оператор (кроме случая использования вложенных операторов без begin-end);
- (j) количество пробелов перед строкой программы должно соответствовать уровню вложенности (по два пробела на уровень вложенности);
- (k) слова begin и end, соответствующие друг другу, должны располагаться строго с одной и той же позиции по вертикали;
- (l) количество строк в подпрограмме и в самой программе (между begin и end) должно быть не более 25 строк;
- (m) три и более сходных по назначению подпрограммы должны выделяться в отдельный модуль;
- (n) в подпрограммах должны отсутствовать случаи использования глобальных переменных;
- (o) в подпрограммах, предназначенных для вычислений не должно быть ввода-вывода; в подпрограммах, предназначенных для ввода-вывода, не должно быть вычислений, кроме тех, что нужны для красивого/корректного ввода-вывода;
- (p) в программе не должны использоваться операторы goto, break, continue; процедуры halt и exit (и их аналоги);
- (q) вместо явно указанных значений чисел, в тексте программы должны использоваться константы.

### 2. Общие требования к надежности

- (a) при работе с файлами должны проверяться все ошибки ввода/вывода;
- (b) при вводе с клавиатуры должна проверяться корректность входных данных (в том числе пользователь может вместо числа ввести строку, число может превышать интервал представления типами данных языка);
- (c) подпрограммы и модули при условии соблюдения всех требований, описанных в комментариях, должны правильно выполняться.

### 3. Общие требования к эргономике

- (a) У всех форм должен быть указано корректное значение поля Title. С помощью Label должны быть описаны все вводимые данные, а также общее назначение формы и ее частей

#### 4. Требования к оптимальности

- (a) Программа и подпрограммы должны быть оптимальны по быстродействию и/или памяти (в случае, если это не требует очень больших усилий).
- (b) В программе и модулях не должно быть набора более двух операторов, который повторяется (в том числе с шаблонными изменениями) в разных местах программы.

### 3 Практическая работа №1. Простые программы с использованием событийно-ориентированного программирования (2 часа)

1. Цель работы: практическое закрепление знаний компонентов Delphi (TForm, TLabel, TEdit, TComboBox, TButton и, возможно, некоторых других – на выбор студента); практическое закрепление знаний об использовании событий в программировании; практическое закрепление знаний о функциях IntToStr; StrToInt; FloatToStr; StrToFloat; исключениях.
2. Порядок выполнения:
- (a) осуществите визуальное проектирование пользовательского интерфейса формы;
- (b) разработайте методы – обработчики тех событий, что необходимо использовать в вашей программе;
- (c) приведите вашу программу в соответствие с требованиями.

Варианты заданий:

#### 0 уровень

- ПП1-0-1 Программа решения квадратного уравнения
- ПП1-0-2 Программа поиска дня недели по числу и месяцу в текущем году
- ПП1-0-3 Программа поиска определителя матрицы  $2 \times 2$
- ПП1-0-4 Программа перевода числа из 10-ой в 16-ую, 8-ую и 2-ую систем.
- ПП1-0-5 Программа поиска времени, когда окончится интервал. Дано: часы и минуты начала интервала и количество минут, сколько он идет. Результат: часы и минуты окончания интервала.
- ПП1-0-6 Программа поиска обратной матрицы для матрицы  $2 \times 2$ .
- ПП1-0-7 Программа поиска длины интервала. Дано: часы и минуты начала интервала и часы и минуты конца интервала. Результат: количество минут в интервале.
- ПП1-0-8 Программа умножения и деления двух комплексных чисел.
- ПП1-0-9 Программа перевода числа из 16-ой, 8-ой и 2-ой системы в 10-ую систему счисления.
- ПП1-0-10 Программа нахождения количества денег на вкладе после окончания его срока по начальному взносу, проценту и срока в годах.
- ПП1-0-11 Программа нахождения степени комплексного числа. Исходные данные: действительная, мнимая часть числа и степень. Результат: действительная и мнимая часть результата.
- ПП1-0-12 Программа умножения и деления чисел, представленных в виде обыкновенных дробей (состоящих из целой части, числителя и знаменателя).
- ПП1-0-13 Программа перевода чисел в/из римской системы счисления.
- ПП1-0-14 Программа определения по дате (число и месяц) знака зодиака.
- ПП1-0-15 Программа определения по обыкновенной дроби (числителю и знаменателю) периода десятичной дроби.
- ПП1-0-16 Программа перевода комплексного числа из обычной формы в тригонометрическую и наоборот.
- ПП1-0-17 Программа-игра Баше. При реализации этого задания не требуется ничего рисовать, вся информация вводится и выводится в виде чисел в обычные элементы управления.
- ПП1-0-18 Программа разложения числа на простые множители.
- ПП1-0-19 Программа нахождения наибольшего общего делителя и наименьшего общего кратного двух натуральных чисел.

ПП1-0-20 Программа-тест по предмету «Прикладное программирование». Создайте программу-тест из 10 вопросов с выбором вариантов ответов и показом результатов прохождения теста.

### 1 уровень

- ПП1-1-1 Программа-календарь (по году, месяцу, числу – день недели) с удобным вводом (с выпадающим списком).
- ПП1-1-2 Программа нахождения решения неравенства  $ax + b$  знак 0 (удобный ввод с выпадающим списком знака неравенства).
- ПП1-1-3 Программа нахождения решения неравенства  $ax^2 + bx + c$  знак 0 (удобный ввод с выпадающим списком знака неравенства).
- ПП1-1-4 Программа учета расходов и доходов по месяцам: в элементы управления вводятся доходы за январь-декабрь и расходы за январь-декабрь, программа выводит прибыль за январь-декабрь, общую сумму расходов, доходов и прибыли за год. Интерфейс должен напоминать таблицу.
- ПП1-1-5 Программа поиска обратной матрицы для матрицы  $3 \times 3$ .
- ПП1-1-6 Программа для изучения ассемблера. Программа должна позволять вводить два двоичных 32-битных числа и выполнять операции ADD, SUB и показывать флаги: AF, CF, SF, PF, ZF, OF
- ПП1-1-7 Программа нахождения центра и радиуса окружности по координатам трех точек, принадлежащих ей.
- ПП1-1-8 Напишите игру «Пятнашки».
- ПП1-1-9 Программа решения системы уравнений

$$\begin{cases} ax + by + c = 0 \\ dx + ey + f = 0 \end{cases}$$

где все представленные числа комплексные. Программа должна находить ответ в случае, если решение единственно, в противном случае выдавать диагностическое сообщение о невозможности решения.

- ПП1-1-10 Программа перевода даты (год, месяцу, число) из григорианского календаря в юлианский и обратно с удобным вводом (с выпадающим списком).
- ПП1-1-11 Программа нахождения площади меньшей из частей круга, порождаемых пересечением круга и прямой, по координатам центра и радиусу круга и по координатам двух точек прямой.
- ПП1-1-12 Программа деления двух многочленов: не более, чем четвертой степени, на не более, чем второй.
- ПП1-1-13 Программа-игра чет-нечет, где человек задумывает, а программа отгадывает четность/нечетность загаданных чисел. Программа должна анализировать загадываемые человеком числа с целью предсказания, выявляя закономерности длиной от 1 до 10 чисел: в случае если человек регулярно загадывает одинаковую последовательность чисел такой длины, то программа с некоторого момента должна их угадывать всегда. Если человек чаще всего после некой последовательности длины от 1 до 10 чисел загадывает чет (или нечет), то программа данное явление также должна выявлять.
- ПП1-1-14 Программа решения уравнения  $ax^2 + bx + c = 0$ , где все представленные числа – комплексные.
- ПП1-1-15 Программа-калькулятор для матриц  $2 \times 2$ . Пользователю доступно задание матриц  $2 \times 2$  и кнопки +, -, \*, /, =, C

### 1.5 уровень

- ПП1-1.5-1 Напишите аналог игры 2048.
- ПП1-1.5-2 Программа нахождения площади пересечения двух кругов, заданных координатами центров и радиусами.
- ПП1-1.5-3 Пусть по некоторым координатам на плоскости установлены столбики. Пользователь окружает веревкой пространство, где расположены эти столбики и максимально ее стягивает. В результате веревка образует многоугольник. Определите ее вершины по координатам расположения столбиков. Столбиков не более 10. Указание: обсуждение этой задачи есть в книге Шеня.
- ПП1-1.5-4 Программа-калькулятор /без использования встроенного компонента/. Напишите аналог программы-калькулятора, входящего в Windows, работающего только в обычном (неинженерном) режиме
- ПП1-1.5-5 Программа-калькулятор /без использования встроенного компонента/. Напишите аналог программы-калькулятора, входящего в Windows, работающего только в обычном (неинженерном) режиме, осуществляющего ввод-вывод в **16-ой системе счисления**. Калькулятор работает только с целыми числами, не требуется реализовывать корень, нахождение обратного числа и проценты. Деление работать должно как целочисленное (div)

- ПП1-1.5-6 Программа-игра в обычные крестики-нолики с компьютером.
- ПП1-1.5-7 Программа нахождения площади пересечения любого числа треугольников (до 100). Вводятся координаты вершин треугольников, вывести площадь многоугольника, который получится в пересечении. Не забудьте сделать проверку на то, что исходные треугольники – это треугольники.

## 2 уровень

- ПП1-2-1 Программа приведения матрицы  $3 \times 3$  к главному ступенчатому виду по шагам (пользователь вводит матрицу и нажимая кнопку много раз последовательно видит процесс приведения матрицы к главному ступенчатому виду)
- ПП1-2-2 Программа нахождения пересечения двух многоугольников. Вводятся координаты вершин двух выпуклых многоугольников (до десятиугольников включительно), вывести координаты вершин и площадь многоугольника, который получится в пересечении. Не забудьте сделать проверку на то, что исходные многоугольники выпуклые. Относительно алгоритма проверка многоугольника на выпуклость смотрите статью на Habrahabr: <http://habrahabr.ru/post/147691/>.
- ПП1-2-3 Программа-калькулятор /без использования встроенного компонента/. Напишите аналог программы-калькулятора, входящего в Windows, работающего только в обычном (неинженерном) режиме, работающий с обыкновенными дробями (состоящими из знака, целой части, числителя и знаменателя).
- ПП1-2-4 Программа обучения делению столбиком. Пользователь вводит два числа и на экран выводится то как выглядит само деление столбиком (можно считать, что количество промежуточных вычитаний не превышает 10).
- ПП1-2-5 Программа обучения переводу из 10-ой в другие системы счисления (от 2-ичной до 36-ричной). Нужно для введенного числа и основания системы счисления вывести то, как выглядит процесс перевода (как обучали в школе – с промежуточным делением столбиком). Можно считать, что количество вспомогательных Label'ов не превышает 50 (их можно создать заранее).
- ПП1-2-6 Программа перевода выражений из обычной формы в обратную польскую форму записи. Допустимы арифметические операции, скобки. В обычной форме действуют обычные правила приоритета. В выражениях используются названия переменных (правила как в Паскале) и числовые константы (правила как в Паскале для типов integer и real).
- ПП1-2-7 Программа обучения делению столбиком многочленов. Пользователь вводит два многочлена и на экран выводится то как выглядит само деление столбиком (можно считать, что количество промежуточных вычитаний не превышает 10).

## 2.5 уровень

- ПП1-2.5-1 Программа-калькулятор с комплексными числами. Напишите аналог программы-калькулятора, входящего в Windows, работающего только в обычном (неинженерном) режиме.
- ПП1-2.5-2 Программа-игра с компьютером в крестики-нолики на поле  $4 \times 4$ . Рекомендуем также почитать статью: <http://www.iqfun.ru/articles/tic-tac-toe-silverman.shtml>
- ПП1-2.5-3 Напишите программу нахождения корней многочлена 3-ей степени. Многочлен и корни над полем комплексных чисел. Надо изучить формулу Кардано.
- ПП1-2.5-4 Программа разложения данного натурального числа в произведение неприводимых гауссовых чисел.
- ПП1-2.5-5 Программа нахождения квадратного корня из целого числа в пространстве  $p$ -адических чисел.
- ПП1-2.5-6 Программа-игра с компьютером в крестики-нолики на поле  $5 \times 5$ . Рекомендуем также почитать статью: <http://www.iqfun.ru/articles/tic-tac-toe-silverman.shtml>
- ПП1-2.5-7 Программа обучения нахождения наибольшего общего делителя двух многочленов (надо учить собственно делению и вычислению дальше – пока не получится наибольший общий делитель).

## 3 уровень

- ПП1-3-1 Напишите программу нахождения корней многочлена 4-ой степени. Многочлен и корни над полем комплексных чисел. Надо изучить формулу Феррари.
- ПП1-3-2 Напишите программу нахождения корней многочлена до 10-ой степени включительно (над полем действительных чисел) в действительных числах. Для решения этой задачи необходимо использовать следующие темы: ряд Штурма, деление многочленов, метод аналогичный методу дихотомии для уточнения корней.

ПП1-3-3 Программа решения системы трех линейных уравнений с тремя неизвестными. Результатом решения должно быть полное описание множества решений системы (даже если их бесконечно много).

3. Содержание отчета:

- (a) Исходный текст программы
- (b) Ответ на контрольные вопросы

4. Контрольные вопросы:

- (a) Какие Вы использовали компоненты в программе, объясните их назначение.
- (b) Что такое исключение? Как использовать это понятие в практическом программировании?
- (c) Что такое событие? Как это понятие используется в событийно-ориентированном программировании?

## 4 Практическая работа №2. Программирование графики и таймера (6 часов)

1. Цель работы: практическое изучение компонентов TImage, TChart, TTimer, TOpenDialog; класса TCanvas.

2. Порядок выполнения:

- (a) осуществите визуальное проектирование пользовательского интерфейса формы (1-ое задание);
- (b) разработайте методы – обработчики тех событий, что необходимо использовать в вашей программе (1-ое задание);
- (c) приведите вашу программу в соответствие с требованиями (1-ое задание).
- (d) осуществите визуальное проектирование пользовательского интерфейса формы (2-ое задание);
- (e) разработайте методы – обработчики тех событий, что необходимо использовать в вашей программе (2-ое задание);
- (f) приведите вашу программу в соответствие с требованиями (2-ое задание).

Второе задание для всех студентов имеет нулевой уровень.

Варианты заданий:

### 1 задание

#### 0 уровень

Идея решения заданий на кинематику состоит в том, что время разбивается на «кванты» (небольшие промежутки), на каждом промежутке времени полагается (приблизительно), что движение является равноускоренным. При расчёте движения на каждом кванте времени вычисляется равнодействующая сил, далее применяется второй закон Ньютона, и далее вычисляется координата и скорость движущегося тела в конце кванта времени.

Задания на геометрическую оптику решаются просто путем нахождения точек пересечения с ближайшей преградой и постепенного рисования линий (пока эта точка не будет достигнута).

ПП2.1-0-1 Пользователь выбирает начальную скорость мяча, направление движения мяча, массу, а программа показывает замедленный «мультифильм» того, как мячик летит с учетом упругого отскока от поверхности (Земли) и сопротивления воздуха ( $\vec{F}_c = -\alpha \vec{v}$ ).  $\alpha$  так же указывается в элементе управления

ПП2.1-0-2 Пользователь выбирает начальную скорость мяча, направление движения мяча (на бильярдном столе), массу, а программа показывает замедленный «мультифильм» того, как мячик летит с учетом упругого отскока от границ бильярдного стола и силы сопротивления ( $\vec{F}_c = -\alpha \vec{v}$ ).  $\alpha$  так же указывается в элементе управления. Лунки учитывать не требуется.

ПП2.1-0-3 Пользователь выбирает начальное направление луча света и круг. После чего программа показывает (в замедленном режиме) распространение этого луча света в предположении, что границы рисунка – идеальные зеркала, а круг состоит из полностью поглощающего свет материала.

ПП2.1-0-4 Пользователь выбирает начальное направление двух лучей света и месторасположение (координаты) обоих источников света. После чего программа показывает (в замедленном режиме) распространение этих лучей света в предположении, что граница рисунка – это идеально отражающие зеркала.



- ПП2.1-0-5 Пользователь выбирает начальное направление луча света и прямоугольник, у которого стороны параллельны осям. После чего программа показывает (в замедленном режиме) распространение этого луча света в предположении, что границы рисунка – идеальные зеркала, а прямоугольник состоит из полностью поглощающего свет материала.
- ПП2.1.0-6 Пользователь выбирает начальное направление луча света и треугольник (координатами вершин). После чего программа показывает (в замедленном режиме) распространение этого луча света в предположении, что границы рисунка – идеальные зеркала, а треугольник состоит из полностью поглощающего свет материала.
- ПП2.1-0-7 Пользователь управляет шаром (для этого нужно использовать события onKeyDown), который перемещается внутри прямоугольной области и не может укатиться за пределы прямоугольной области. Начальное положение и размер шара задается в элементах управления TEdit.
- ПП2.1-0-8 Пользователь управляет рисованием двумя линиями внутри прямоугольной области (непрерывными), используя клавиши-стрелки и TAB (для переключения между линиями). Первые точки линий задаются координатами в TEdit.
- ПП2.1-0-9 Пользователь выбирает радиусы двух дисков и скорость вращения одного из них. Программа рисует эти два диска, соединенные ременной передачей. Для отображения вращения на каждом диске надо рисовать два перпендикулярных диаметра.
- ПП2.1-0-10 Нарисуйте тележку, которая передвигается между двумя краями рисунка с правильным отображением вращения ее колес. Скорость движения выбирается пользователем. При достижении края рисунка скорость мгновенно меняется на противоположно направленную.
- ПП2.1-0-11 Нарисуйте следующее движение. Нить зацеплена за зубчик вращающегося диска, эта нить перекинута через блок и удерживает прямоугольный объект. Скорость вращения, радиус диска задается пользователем.
- ПП2.1-0-12 Нарисуйте следующее движение. Шар с заданной начальной скоростью и радиусом катится по горизонтальной поверхности, при ударе о края рисунка его скорость меняется на противоположную. При этом скорость его вращения замедляется (скорость на каждом кванте времени умножается на вводимый коэффициент  $k$ ). На шаре надо нарисовать пару перпендикулярных линий для отображения вращения.
- ПП2.1-0-13 **Моделирование движения спутника.** Нарисуйте следующее движение. Один шар неподвижен, другой шар имеет начальную скорость (вектор, задается пользователем), при этом на него действует сила в направлении первого шара, имеющая величину  $\frac{k}{r^2}$ , где  $k$  вводится пользователем, а  $r$  – расстояние от шара до неподвижного шара. Движение прекращается в случае, если шары соприкасаются. Движение рассчитывается на каждом кванте времени исходя из того, что оно в течении кванта равноускоренное и прямолинейное.
- ПП2.1-0-14 Нарисуйте следующее движение. Шар движется по поверхности, расположенной к горизонту под углом  $\alpha$  (вводится пользователем), и изначально расположен сверху. При ударе о край рисунка движение заканчивается. При этом на него действует сила тяжести (масса также вводится). На шаре надо нарисовать пару перпендикулярных линий для отображения вращения.
- ПП2.1-0-15 Нарисуйте следующее движение. Один диск расположен неподвижно, другой вращается, при этом (за счет сцепления) вращается вокруг первого диска. Пользователем задаются радиусы дисков и скорость вращения вокруг своей оси второго из них. На дисках надо нарисовать по два перпендикулярных отрезка для отслеживания вращения.
- ПП2.1-0-16 Нарисуйте следующее движение. Шар движется по поверхности, расположенной к горизонту под углом  $\alpha$  (вводится пользователем), и изначально расположен внизу. При ударе о край рисунка движение заканчивается. При этом на него действует сила тяжести (масса также вводится). Начальная скорость задается пользователем (она направлена параллельно поверхности вверх). На шаре надо нарисовать пару перпендикулярных линий для отображения вращения.
- ПП2.1-0-17 Пользователь выбирает начальную скорость мяча, направление движения мяча (на бильярдном столе), массу, а программа показывает замедленный «мультифильм» того, как мячик летит с учетом упругого отскока от границ бильярдного стола. Если при этом мяч попадает в квадрат данного размера вблизи углов бильярдного стола, то он исчезает (проваливается в лунку).
- ПП2.1-0-18 На одной горизонтальной прямой расположены центры двух шаров, которые движутся в направлении этой прямой со скоростью, задаваемой пользователем. Массы шаров так же задаются, радиусы одинаковы. При достижении границ рисунка шары упруго отскакивают; аналогично упруго отскакивают они и друг от друга. Нарисуйте такое движение.

- ПП2.1-0-19 На одной горизонтальной прямой расположены центры двух шаров одинаковой массы, но разных радиусов (задаются пользователем). Шары движутся в направлении этой прямой со скоростями, задаваемыми пользователем. При достижении границ рисунка шары упруго отскакивают; аналогично упруго отскакивают они и друг от друга. Нарисуйте такое движение.
- ПП2.1-0-20 Рисунок по горизонтали разделен на две одинаковые по размеру части. Предполагается, что сверху показатель преломления  $\alpha$ , снизу –  $\beta$ . Границы рисунка – идеальные зеркала. Пользователь выбирает место источника луча и начальное направление луча света, после чего программа показывает (в замедленном режиме) распространение этого луча света.
- ПП2.1-0-21 Рисунок по горизонтали разделен на три одинаковые по размеру части. Предполагается, что сверху и снизу показатель преломления  $\alpha$ , а посередине среда – это вакуум. Границы рисунка – идеальные зеркала. Пользователь выбирает место источника луча и начальное направление луча света, после чего программа показывает (в замедленном режиме) распространение этого луча света.
- ПП2.1-0-22 Рисунок по горизонтали разделен на две одинаковые по размеру части. При этом левая половина разделительной линии – зеркальная, а правая – прозрачная. Предполагается, что сверху показатель преломления  $\alpha$ , снизу среда – вакуум. Границы рисунка – идеальные зеркала. Пользователь выбирает место источника луча и начальное направление луча света, после чего программа показывает (в замедленном режиме) распространение этого луча света.
- ПП2.1-0-23 На одной горизонтальной прямой расположены центры двух кубов, которые движутся в направлении этой прямой со скоростью, задаваемой пользователем (сами кубы ориентированы так, что стороны параллельны осям). Массы кубов так же задаются, размеры одинаковы. При достижении границ рисунка кубы упруго отскакивают; аналогично упруго отскакивают они и друг от друга. Нарисуйте такое движение.

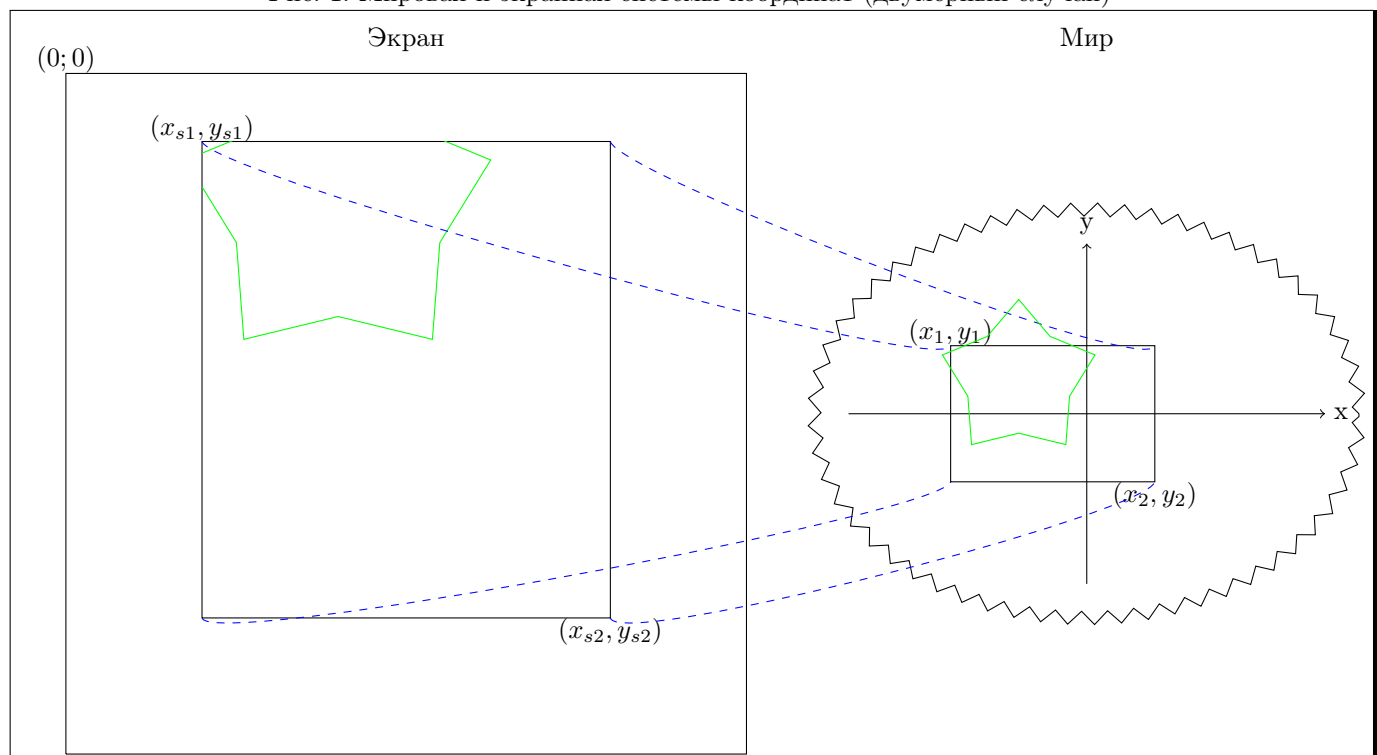
## 1 уровень

- ПП2.1-1-1 Пользователь выбирает начальную скорость двух мячей, направление движения мячей (на бильярдном столе), а программа показывает замедленный «мультифильм» того, как мячи летят с учетом упругого отскока от границ бильярдного стола и при столкновении друг с другом, а также силы сопротивления ( $\vec{F}_c = -\alpha \vec{v}$ ).  $\alpha$  так же указывается в элементе управления. Лунки учитывать не требуется.
- ПП2.1-1-2 Пользователь выбирает координаты расположения и направление испускания источника света и координаты левого верхнего и правого нижнего угла прямоугольника, соответствующего емкости с веществом, у которого показатель преломления  $\alpha$  (также вводится). Предполагается, что границы рисунка – это идеально отражающие зеркала. Покажите, как будет идти луч света (в чрезвычайно замедленной «съемке») от источника (след луча света на рисунке остается).
- ПП2.1-1-3 Пользователь выбирает координаты расположения источника света (направление луча перпендикулярно линзе) и координаты начала и конца отрезка, символизирующего линзу с фокусным расстоянием  $f$  (также вводится). Предполагается, что границы рисунка – это идеально отражающие зеркала. Покажите, как будет идти луч света (в чрезвычайно замедленной «съемке») от источника (след луча света на рисунке остается).
- ПП2.1-1-4 Пользователь выбирает начальную скорость двух мячей, направление движения мячей (на бильярдном столе), а программа показывает замедленный «мультифильм» того, как мячи летят с учетом упругого отскока от границ бильярдного стола и при столкновении друг с другом. Кроме того указывается прямоугольник координатами левого верхнего и правого нижнего угла, от которого шарики тоже отскакивают. Лунки учитывать не требуется.
- ПП2.1-1-5 Пользователь выбирает координаты расположения и направление испускания источника света и координаты вершин треугольника, соответствующего емкости с веществом, у которого показатель преломления  $\alpha$  (также вводится). Предполагается, что границы рисунка – это идеально отражающие зеркала. Покажите, как будет идти луч света (в чрезвычайно замедленной «съемке») от источника (след луча света на рисунке остается).
- ПП2.1-1-6 Пользователь выбирает начальную скорость двух мячей, направление движения мячей (на бильярдном столе), а программа показывает замедленный «мультифильм» того, как мячи летят с учетом упругого отскока от границ бильярдного стола и при столкновении друг с другом. Кроме того указывается треугольник координатами вершин, от которого шарики тоже отскакивают. Лунки учитывать не требуется.
- ПП2.1-1-7 Пользователь выбирает начальную скорость мяча, направление движения мяча (на бильярдном столе), а программа показывает замедленный «мультифильм» того, как мяч летит с учетом упругого

столкновения с границами стола, а также с движущимся параллелепипедом (указывается левый верхний и правый нижний угол параллелепипеда, его скорость). Параллелепипед движется слева направо, а, после достижения границы стола, справа налево и так далее.

- ПП2.1-1-8 Пользователь выбирает начальную скорость мяча, направление движения мяча (на бильярдном столе), а программа показывает замедленный «мультфильм» того, как мяч летит с учетом упругого столкновения с границами стола, а также с движущимся параллелепипедом (указывается левый верхний и правый нижний угол параллелепипеда, его скорость). Параллелепипед движется сверху вниз, а, после достижения границы стола, снизу вверх и так далее.
- ПП2.1-1-9 Пользователь выбирает координаты расположения и направление испускания источника света и координаты концов отрезков, соответствующего идеальному зеркалу. Предполагается, что границы рисунка – это также идеально отражающие зеркала. Покажите, как будет идти луч света (в чрезвычайно замедленной «съемке») от источника (след луча света на рисунке остается).
- ПП2.1-1-10 Пользователь выбирает начальную скорость шайбы и скорость ее вращения, а программа показывает ее движение на прямоугольном поле с учетом упругого отталкивания от границы поля. При этом предполагается, что сила сопротивления, действующая на шайбу, пропорциональна скорости движения/скорости вращения. На шайбе необходимо рисовать два перпендикулярных отрезка для отображения вращения.
- ПП2.1-1-11 Пользователь выбирает радиусы двух дисков и ширину зубцов и расстояние между ними (общую для каждого из шаров). Программа должна нарисовать эти два диска и цепь, их соединяющую. При этом первый диск должен вращаться с заданной пользователем скоростью.
- ПП2.1-1-12 Пользователь выбирает радиусы двух дисков и расположение зубца на каждом из них. Зубцы соединены перекинутой через блок натянутой ниткой. Первый диск вращается с заданной пользователем скоростью в том интервале углов, в котором нить не обрывается. При достижении границы интервала скорость мгновенно меняется. Если нить не обрывается при любом положении первого диска, то он просто вращается.
- ПП2.1-1-13 **Моделирование движения двух планет.** Нарисуйте следующее движение. Два шара имеют начальную скорость (вектора, задаются пользователем), при этом на них действуют силы в направлении другого шара, имеющая величину  $\frac{k}{r^2}$ , где  $k$  вводится пользователем и зависит от шара, а  $r$  – расстояние от шара до другого шара. Движение прекращается в случае, если шары соприкасаются. Движение рассчитывается на каждом кванте времени исходя из того, что оно в течении кванта равноускоренное и прямолинейное.
- ПП2.1-1-14 **Моделирование движения спутника.** Нарисуйте следующее движение. Один шар неподвижен, другой шар имеет начальную скорость (вектор, задается пользователем), при этом на него действует сила в направлении первого шара, имеющая величину  $\frac{k}{r^2}$ , где  $k$  вводится пользователем, а  $r$  – расстояние от шара до неподвижного шара. Движение прекращается в случае, если шары соприкасаются. При этом, если пользователь нажимает клавишу пробел, то спутнику сообщается дополнительное ускорение в направлении его движения (величина ускорения вводится пользователем предварительно). Движение рассчитывается на каждом кванте времени исходя из того, что оно в течении кванта равноускоренное и прямолинейное.
- ПП2.1-1-15 **Моделирование движения двух спутников.** Нарисуйте следующее движение. Два шара имеют начальную скорость и координаты (скорость – это вектор, задаются пользователем), при этом на них действуют силы в направлении третьего шара, имеющие величину  $\frac{k}{r^2}$ , где  $k$  вводится пользователем и зависит от шара, а  $r$  – расстояние от шара до другого шара. Движение прекращается в случае, если какие-либо шары соприкасаются. Движение рассчитывается на каждом кванте времени исходя из того, что оно в течении кванта равноускоренное и прямолинейное.
- ПП2.1-1-16 Нарисуйте следующее движение. Нарисованы две поверхности, которые соприкасаются внизу рисунка, расположенные к горизонту под углом  $\alpha$  и  $\beta$  соответственно (углы вводятся). Шар изначально расположен сверху и двигается под силой тяжести и силой сопротивления воздуха (масса также вводится). Сила сопротивления воздуха учитывается следующим образом: на каждом кванте времени скорость умножается на коэффициент  $k$  (вводимый пользователем). На шаре надо нарисовать пару перпендикулярных линий для отображения вращения.
- ПП2.1-1-17 Пользователь выбирает начальную скорость двух мячей, направление движения мячей (на бильярдном столе), а программа показывает замедленный «мультфильм» того, как мячи летят с учетом упругого отскока от границ бильярдного стола и при столкновении друг с другом. Кроме того указывается выпуклый четырехугольник координатами вершин, от которого шарики тоже отскакивают. Лунки учитывать не требуется.

Рис. 1: Мировая и экранная системы координат (двумерный случай)



## 2 уровень

### Теоретическое введение

Все современные программы, работающие в сфере компьютерной графики, работают на основе следующих представлений.

На экране (или части экрана) отображается часть «мира». Миром может быть игровой мир в играх, физическая модель в задачах моделирования, плоскость или трехмерное пространство в математических задачах. В любом случае в «мире» вводится своя система координат (**мировая**), а экрану (части экрана) присуща экранная система координат.

Два основных действия в таком подходе – это нахождение соответствующих координат на экране для точек мира и нахождение точек мира, соответствующих точке экрана (в некоторых случаях таких точек в мире много).

Мы рассмотрим простейший случай: когда в мире введена двумерная декартова система координат. И приведем формулы перевода из мировых в экранную систему координат и наоборот. Ниже для заданий второго уровня будут также указаны формулы перевода из мировых в экранную систему координат в трехмерном случае.

Итак, пусть мы хотим отобразить на экран (часть экрана) прямоугольный фрагмент мира со сторонами, параллельными осям координат с противоположными вершинами  $(x_1, y_1)$  и  $(x_2, y_2)$  (см. рис. 1). Часть экрана, на которую мы отображаем фрагмент мира – аналогичный прямоугольник с вершинами  $(x_{s1}, y_{s1})$  и  $(x_{s2}, y_{s2})$ . Обозначим  $(x, y)$  – координаты точки в мире, а  $x_s, y_s$  – координаты соответствующей точки на экране. Тогда

$$\begin{aligned}
 x_s &= \left[ (x - x_1) \frac{x_{s2} - x_{s1}}{x_2 - x_1} + x_{s1} \right] \\
 y_s &= \left[ (y - y_1) \frac{y_{s2} - y_{s1}}{y_2 - y_1} + y_{s1} \right] \\
 x &= (x_s - x_{s1}) \frac{x_2 - x_1}{x_{s2} - x_{s1}} + x_1 \\
 y &= (y_s - y_{s1}) \frac{y_2 - y_1}{y_{s2} - y_{s1}} + y_1
 \end{aligned} \tag{1}$$

Здесь [...] – это округление до целого.

На практике часто приходится производить преобразования объектов находящихся в «мире», например перемещать и вращать фигуры. Реализуются такие действия над отдельными точками, которые впоследствии используются для рисования тех или иных примитивов (треугольников, отрезков, тетраэдров).

Перемещение точки осуществляется с использованием простейшего преобразования:

$$\begin{aligned}x &\rightarrow x + dx \\y &\rightarrow y + dy\end{aligned}$$

Здесь  $(dx, dy)$  – вектор, на который смещаются все точки фигуры.

Вращение фигуры вокруг точки  $(0; 0)$  осуществляется путем умножения координат точек на матрицу поворота на угол  $\varphi$  против часовой стрелки:

$$\begin{pmatrix} x \\ y \end{pmatrix} \rightarrow \begin{pmatrix} \cos \varphi & -\sin \varphi \\ \sin \varphi & \cos \varphi \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix}$$

Вращение фигуры вокруг произвольной точки  $(x_0; y_0)$  производится тремя преобразованиями подряд: сдвиг на  $(-x_0; -y_0)$ , вращение и сдвиг на  $(x_0; y_0)$ .

Существует способ, который позволяет делать вращение, сдвиг фигур и любые комбинации таких преобразований с помощью одного и того же математического действия: умножения на матрицу. Для этого вместо

обычных координат точки  $\begin{pmatrix} x \\ y \end{pmatrix}$  рассматривают её однородные координаты  $\begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$ , то есть добавляют

фиктивную третью координату, которая всегда равна 1. В этом случае поворот на угол  $\varphi$  осуществляется умножением на матрицу

$$\begin{pmatrix} \cos \varphi & -\sin \varphi & 0 \\ \sin \varphi & \cos \varphi & 0 \\ 0 & 0 & 1 \end{pmatrix},$$

а сдвиг на вектор  $(dx, dy)$  – на матрицу

$$\begin{pmatrix} 1 & 0 & dx \\ 0 & 1 & dy \\ 0 & 0 & 1 \end{pmatrix},$$

Любой комбинации поворотов и сдвигов (то есть любому движению) соответствует умножение на матрицу – произведение соответствующих матриц поворотов и сдвигов, записанных в обратном порядке.

При использовании однородных координат преобразование мировых координат точки в экранные также можно осуществлять путем умножения на матрицу (с последующим округлением):

$$\begin{pmatrix} \frac{x_{s2}-x_{s1}}{x_2-x_1} & 0 & x_{s1}-x_1 & \frac{x_{s2}-x_{s1}}{x_2-x_1} \\ 0 & \frac{y_{s2}-y_{s1}}{y_2-y_1} & y_{s1}-y_1 & \frac{y_{s2}-y_{s1}}{y_2-y_1} \end{pmatrix}$$

### Теоретическое введение для заданий 3 уровня

Проведем модификацию предыдущей теории для случая трехмерной графики. В этом случае преобразо-

вания осуществляется с точками, имеющими четыре однородные координаты:  $\begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix}$ .

Любое движение фигуры в трехмерном случае можно представить в виде комбинации вращений вокруг каждой из осей  $(x, y$  и  $z)$  и сдвигов.

Матрица поворота на угол  $\phi$  вокруг оси  $x$ :

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos \phi & -\sin \phi & 0 \\ 0 & \sin \phi & \cos \phi & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Матрица поворота на угол  $\psi$  вокруг оси  $y$ :

$$\begin{pmatrix} \cos \psi & 0 & -\sin \psi & 0 \\ 0 & 1 & 0 & 0 \\ \sin \psi & 0 & \cos \psi & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Матрица поворота на угол  $\theta$  вокруг оси  $z$ :

$$\begin{pmatrix} \cos \theta & -\sin \theta & 0 & 0 \\ \sin \theta & \cos \theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Матрица сдвига на вектор  $(a; b; c)$ :

$$\begin{pmatrix} 1 & 0 & 0 & a \\ 0 & 1 & 0 & b \\ 0 & 0 & 1 & c \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

При получении координат точки экрана, соответствующей точке в «мире» мы должны каким-то образом избавиться от третьей координаты. Существует два основных способа это сделать: параллельная и центральная проекция. Первая из них используется в инженерной практике, вторая – в большинстве игр.

Матрица отображения на экран (для параллельной проекции):

$$\begin{pmatrix} \frac{x_s2-x_{s1}}{x_2-x_1} & 0 & 0 & x_{s1} - \frac{x_1(x_s2-x_{s1})}{x_2-x_1} \\ 0 & -\frac{y_s2-y_{s1}}{y_2-y_1} & 0 & y_{s1} - \frac{y_1(y_s2-y_{s1})}{y_2-y_1} \end{pmatrix}$$

Отображение на экран для центральной проекции строится следующим образом. Имеется точка  $(x_f; y_f; z_f)$  – фокус (в ней находится «глаз»); имеется плоскость экрана ( $z = z_s$ ). Чтобы найти координаты центральной проекции на плоскости экрана  $(x_s; y_s)$  используются следующие формулы:

$$\begin{cases} x_u = \frac{z_s - z_f}{z - z_f}(x - x_f) + x_f \\ y_u = \frac{z_s - z_f}{z - z_f}(y - y_f) + y_f \end{cases}$$

Для того, чтобы перейти далее перейти к экранным координатам от координат  $(x_u; y_u)$  используются формулы (1) для двумерного случая. В случае использования центральной проекции рекомендую брать экранную плоскость  $z_s = -10$ , а фокус  $(x_f; y_f; z_f) = (0; 0; -20)$ .

#### Собственно задания

- ПП2.1-2-1 Написать программу, которая будет выводить на экран проекцию куба на плоскость. При этом должна быть возможность управлять вращением этой фигуры вокруг оси  $x$ ; оси  $y$ ; оси  $z$  (по отдельности); а также сдвигать ее в разные стороны по всем трем осям
- ПП2.1-2-2 Написать программу, которая будет выводить на экран проекцию правильного тетраэдра на плоскость. При этом должна быть возможность управлять вращением этой фигуры вокруг оси  $x$ ; оси  $y$ ; оси  $z$  (по отдельности); а также сдвигать ее в разные стороны по всем трем осям
- ПП2.1-2-3 Написать программу, которая будет выводить на экран проекцию правильного октаэдра на плоскость. При этом должна быть возможность управлять вращением этой фигуры вокруг оси  $x$ ; оси  $y$ ; оси  $z$  (по отдельности); а также сдвигать ее в разные стороны по всем трем осям
- ПП2.1-2-4 Написать программу, которая будет выводить на экран проекцию правильной прямоугольной пирамиды на плоскость. При этом должна быть возможность управлять вращением этой фигуры вокруг оси  $x$ ; оси  $y$ ; оси  $z$  (по отдельности); а также сдвигать ее в разные стороны по всем трем осям
- ПП2.1-2-5 Написать программу, которая будет выводить на экран центральную проекцию куба на плоскость. При этом должна быть возможность управлять вращением этой фигуры вокруг оси  $x$ ; оси  $y$ ; оси  $z$  (по отдельности); а также сдвигать ее в разные стороны по всем трем осям
- ПП2.1-2-6 Написать программу, которая будет выводить на экран центральную проекцию тетраэдра на плоскость. При этом должна быть возможность управлять вращением этой фигуры вокруг оси  $x$ ; оси  $y$ ; оси  $z$  (по отдельности); а также сдвигать ее в разные стороны по всем трем осям
- ПП2.1-2-7 Написать программу, которая будет выводить на экран центральную проекцию прямоугольной пирамиды на плоскость. При этом должна быть возможность управлять вращением этой фигуры вокруг оси  $x$ ; оси  $y$ ; оси  $z$  (по отдельности); а также сдвигать ее в разные стороны по всем трем осям
- ПП2.1-2-8 Написать программу, которая будет выводить на экран центральную проекцию октаэдра на плоскость. При этом должна быть возможность управлять вращением этой фигуры вокруг оси  $x$ ; оси  $y$ ; оси  $z$  (по отдельности); а также сдвигать ее в разные стороны по всем трем осям

- ПП2.1-2-9 Написать программу, которая будет выводить на экран центральную проекцию икосаэдра на плоскость. При этом должна быть возможность управлять вращением этой фигуры вокруг оси  $x$ ; оси  $y$ ; оси  $z$  (по отдельности); а также сдвигать ее в разные стороны по всем трем осям
- ПП2.1-2-10 Написать программу, которая будет выводить на экран проекцию икосаэдра на плоскость. При этом должна быть возможность управлять вращением этой фигуры вокруг оси  $x$ ; оси  $y$ ; оси  $z$  (по отдельности); а также сдвигать ее в разные стороны по всем трем осям

## 2 задание

Второе задание представлено в работе только нулевым уровнем.

- ПП2.2-0-1 В каждой строке текстового файла указано четыре числа – координаты начала и конца отрезка. Выведите на экран эти отрезки (в TChart).
- ПП2.2-0-2 В каждой строке текстового файла указано два числа – координаты точки. Выведите на экран отрезки, соединяющие данные соседние точки (в TChart).
- ПП2.2-0-3 В каждой строке текстового файла указано два числа – координаты точки в полярных координатах. Выведите на экран отрезки, соединяющие данные соседние точки (в TChart).
- ПП2.2-0-4 В каждой строке текстового файла указано четыре числа – координаты начала и конца отрезка в полярных координатах. Выведите на экран эти отрезки (в TChart).
- ПП2.2-0-5 В каждой строке текстового файла указаны две координаты точки и номера точек (нумерация с единицы, указывается номер строки в файле), с которыми нужно соединить данную точку. Выведите эти точки и соединяющие их отрезки в TChart.
- ПП2.2-0-6 В каждой строке текстового файла указаны две координаты точки в полярных координатах и номера точек (нумерация с единицы, указывается номер строки в файле), с которыми нужно соединить данную точку. Выведите эти точки и соединяющие их отрезки в TChart.
- ПП2.2-0-7 В каждой строке текстового файла расположены координаты двух точек. Необходимо нарисовать стрелку от первой точки ко второй для каждой такой пары точек в TChart.
- ПП2.2-0-8 В первой строке указаны координаты трех точек – вершин треугольника, а в каждой следующей строке – два числа, показывающие на сколько надо сместить этот треугольник по оси абсцисс и ординат соответственно. Выведите все эти треугольники в TChart.
- ПП2.2-0-9 В первой строке указаны координаты трех точек – вершин четырехугольника, а в каждой следующей строке – два числа, показывающие на сколько надо сместить этот четырехугольник по оси абсцисс и ординат соответственно. Выведите все эти треугольники в TChart.
- ПП2.2-0-10 В первой строке указаны координаты трех точек – вершин треугольника, а в каждой следующей строке – по одному числу, показывающему на сколько градусов надо повернуть этот треугольник вокруг начала координат. Выведите все эти треугольники в TChart (исходный и повернутые на указанные углы).
- ПП2.2-0-11 В первой строке указаны координаты начала и конца первого вектора, а в каждой следующей строке указаны координаты вектора, который прибавляется к предыдущему вектору (то есть результату суммирования). Отобразите все эти векторы в TChart.
- ПП2.2-0-12 Выведите в TChart точки, каждая из которых представлена двумя числами в текстовом файле. При этом каждую пару точек соедините отрезком.
- ПП2.2-0-13 В каждой строке файла указаны по два вещественных числа. Первое число – это абсциссы точек, а второе число – это приращение ординаты точек (для первой точки – сама ордината). Выведите в TChart эти точки, последовательно соединенные отрезками. В файле абсциссы точек расположены в порядке возрастания.
- ПП2.2-0-14 В каждой строке файла указаны по два вещественных числа. Первое число – это абсциссы точек, а второе число – это приращение ординаты точек (для первой точки – сама ордината). Выведите в TChart столбцовую диаграмму (TBarSeries) по данным координатам абсцисс и ординат (ординаты – это высоты столбцов). В файле абсциссы точек расположены в порядке возрастания.
- ПП2.2-0-15 В первой строке указаны координаты начала и конца первого вектора, а в каждой следующей строке указаны координаты вектора, который пририсовывается к предыдущему вектору. Отобразите все эти векторы в TChart.
- ПП2.2-0-16 В первой строке указаны абсциссы всех точек, а в каждой следующей строке ординаты точек, которые нужно соединить (второй-последней строкам соответствует по одной линии). Вывод сделать в TChart.

- ПП2.2-0-17 В первой строке указаны радиусы всех точек, а в каждой следующей строке углы для всех точек, которые нужно соединить (второй-последней строкам соответствует по одной линии). Вывод сделать в TChart. Подсказка: вывод надо осуществлять в TPolarSeries.
- ПП2.2-0-18 В первой строке каждой пары строк файла хранятся абсциссы точек, во второй – ординаты. Выведите соединенные отрезками точки из каждой пары строк файла в TChart (количество линий будет равно количеству строку файла, деленному на два).
- ПП2.2-0-19 В каждой строке файла указано по два числа – абсциссы и ординаты точек. Выведите на единый график столбчатую диаграмму и линию, соединяющую данные точки.
- ПП2.2-0-20 В первой строке указаны углы всех точек, а в каждой следующей строке радиусы для всех точек, которые нужно соединить (второй-последней строкам соответствует по одной линии). Вывод сделать в TChart. Подсказка: вывод надо осуществлять в TPolarSeries.
- ПП2.2-0-21 В первой строке каждой пары строк файла хранятся радиусы точек, во второй – углы. Выведите соединенные отрезками точки из каждой пары строк файла в TChart (количество линий будет равно количеству строку файла, деленному на два). Подсказка: необходимо использовать TPolarSeries.
- ПП2.2-0-22 В первой строке файла указаны абсциссы всех точек, во второй – ординаты. Выведите на единый график столбчатую диаграмму и линию, соединяющую данные точки.
- ПП2.2-0-23 В первой строке указаны ординаты всех точек, а в каждой следующей строке абсциссы точек, которые нужно соединить (второй-последней строкам соответствует по одной линии). Вывод сделать в TChart.
- ПП2.2-0-101 Согласно данному файлу выведите диаграмму (ChartBarSeries) частоты использования значений байтов (от 0 до 255)
- ПП2.2-0-102 Согласно данному текстовому файлу выведите диаграмму (ChartBarSeries) длин строк.
- ПП2.2-0-103 В текстовом файле заданы многоугольники координатами вершин (каждому многоугольнику соответствует одна строка текстового файла). Выведите их в TChart.
- ПП2.2-0-104 Согласно данному файлу выведите диаграмму (ChartBarSeries) частоты использования букв латинского алфавита без учета регистра.
- ПП2.2-0-105 Согласно данному текстовому файлу выведите график зависимости среднего значения чисел (вещественных), записанных в строке, от номера строки.
- ПП2.2-0-106 В каждой строке текстового файла указано четыре числа – координаты начала и конца отрезка (в полярной системе координат). Выведите на экран эти отрезки (в TChart).
- ПП2.2-0-107 В каждой строке текстового файла указано два числа – координаты точки в полярной системе координат. Выведите на экран отрезки, соединяющие данные соседние точки (в TChart).
- ПП2.2-0-108 В текстовом файле заданы многоугольники координатами вершин в полярной системе координат (каждому многоугольнику соответствует одна строка текстового файла). Выведите их в TChart.
- ПП2.2-0-109 Согласно данному файлу выведите диаграмму (ChartPieSeries) частоты использования значений байтов (от 0 до 255)
- ПП2.2-0-110 В данном файле заданы вещественные числа. Отобразите их в ChartBarSeries, при этом по оси X отображаются последовательно числа 1, 2, 3 и так далее.
- ПП2.2-0-111 В данном текстовом файле заданы вещественные числа – это углы в градусах. Выведите в TChart отрезки от начала координат под заданным углом длины 1.
- ПП2.2-0-112 Согласно данному файлу выведите диаграмму (ChartBarSeries) частоты использования цифр в файле.
- ПП2.2-0-113 Согласно данному файлу выведите линию, которая соединяет точки с координатами: (значение цифры, количество раз ее использования в файле).
- ПП2.2-0-114 Согласно данному файлу выведите линию, соединяющую точки с координатами (значение байта, количество раз его использования в файле).
- ПП2.2-0-115 В каждой строке текстового файла указывается номер дня данного месяца, цена акции на момент открытия торга и на момент закрытия торга. Выведите отрезки, каждый из которых соединяет две точки, обе из которых имеют одинаковую абсциссу – номер дня, а ординаты – это цена акции на моменты открытия и закрытия.
- ПП2.2-0-116 В каждой строке текстового файла указывается номер дня данного месяца, цена акции на момент открытия торга и на момент закрытия торга. Соедините точки с координатами (номер дня, среднее арифметическое цены акции на моменты открытия и закрытия в данный день).
- ПП2.2-0-117 В каждой строке текстового файла указывается номер дня данного месяца, цена акции на момент открытия торга и на момент закрытия торга. Соедините точки с координатами (номер дня, цены акции на моменты открытия) и другой линией точки (номер дня, цена на момент закрытия)



- ПП2.2-0-201 Согласно данному текстовому файлу выведите диаграмму (ChartPieSeries) длин строк.
- ПП2.2-0-202 Согласно данному файлу выведите диаграмму (ChartPieSeries) частоты использования букв латинского алфавита без учета регистра
- ПП2.2-0-203 В первой и единственной строке текстового файла указано три числа – координаты точки и радиус окружности. Выведите ее в TChart (ее надо изображать малыми отрезками)
- ПП2.2-0-204 В первой и единственной строке текстового файла указано три числа – координаты точки в полярных координатах и радиус окружности. Выведите их в TChart (ее надо изображать малыми отрезками)
- ПП2.2-0-205 В файле записаны вещественные числа. Промежуток от минимального до максимального числа надо разделить на 10 равных участков и вывести диаграмму (ChartBarSeries), показывающую частоту попадания чисел на соответствующий участок (частота – это отношение количества чисел, попавших на участок, к количеству всех чисел). В будущем такой график вы встретите в курсе теории вероятностей (гистограмма)
- ПП2.2-0-206 В файле записаны вещественные числа. Промежуток от минимального до максимального числа надо разделить на 10 равных участков и вывести диаграмму (ChartPieSeries), показывающую частоту попадания чисел на соответствующий участок (частота – это отношение количества чисел, попавших на участок, к количеству всех чисел).
- ПП2.2-0-207 В файле записаны вещественные числа. Выведите в TChart график функции, равной отношению количества чисел, меньших  $x$  к количеству всех чисел. В будущем такой график вы встретите в курсе теории вероятностей (выборочная функция распределения).
- ПП2.2-0-208 В каждой строке текстового файла указано несколько пар чисел (пара чисел соответствует одной точке – декартовы координаты). Для каждой строки выведите в TChart линию, соединяющую последовательно точки.
- ПП2.2-0-209 В каждой строке текстового файла указано несколько пар чисел (пара чисел соответствует одной точке – полярные координаты). Для каждой строки выведите в TChart линию, соединяющую последовательно точки.
- ПП2.2-0-210 В каждой строке текстового файла указано несколько пар чисел (до 10-ти; пара чисел соответствует одной точке – декартовы координаты). Для каждой строки выведите в TChart отрезки, соединяющие все точки друг с другом.

### 3. Содержание отчета:

- Исходный текст первой программы
- Исходный текст второй программы
- Ответы на контрольные вопросы

### 4. Контрольные вопросы:

- Какие средства предоставляет для рисования TCanvas?
- Как практически осуществляется использование TChart для рисования?

## 5 Практическая работа №3. Сложные элементы управления (4 часа)

1. Цель работы: практическое закрепление знаний компонентов TTreeView или TListView.

### 2. Порядок выполнения:

- осуществите визуальное проектирование пользовательского интерфейса формы;
- разработайте методы – обработчики тех событий, что необходимо использовать в вашей программе;
- приведите вашу программу в соответствие с требованиями.

### Нулевой уровень

Обратите внимание, что использование каких-либо функций для работы с файлами, кроме стандартных в этом задании запрещено.

ПП3-0-1 Вывести с помощью TListView статистическую информацию о выбранном текстовом файле: для каждой строки указывается номер строки, первое слово, количество символов в ней, количество слов в ней.

- ППЗ-0-2 Вывести с помощью TTreeView слова из выбранного текстового файла. В верхнем уровне иерархия находятся первые символы, в следующем уровне пары символов слов и так далее. Листья иерархия – это слова, что встречаются в файле
- ППЗ-0-3 Вывести с помощью TListView решения квадратного уравнения, коэффициенты которого указаны в текстовом файле (по три коэффициента на строке): в компоненте должно быть три столбца: само уравнение, первый корень, второй корень. В случае, если уравнение не квадратное в последних двух столбцах информация не выводится.
- ППЗ-0-4 Из текстового файла читаются числа и записываются в сортирующее бинарное дерево. Отобразите его с помощью TTreeView.
- ППЗ-0-5 По содержимому текстового файла, содержащего в каждой строке по три числа, обозначающих длины сторон треугольника, вывести в TListView в отдельных столбцах длины сторон треугольника, его периметр, площадь и информацию о том, является ли он тупоугольным, прямоугольным или остроугольным.
- ППЗ-0-6 В текстовом файле записана иерархически организованная информация подобно следующему примеру:
- \* Простые типы
  - \*\* Целые типы
  - \*\*\* real
  - \*\*\* integer
  - \*\* Вещественные типы
  - \* Указатели
  - \*\* Типизированные
  - \*\* Бестиповые
  - \* Составные типы
  - \*\* массивы
  - \*\* записи
- вывести эту иерархию в TTreeView.
- ППЗ-0-7 Выведите с помощью TListView информацию о разложении чисел в заданном интервале на множители. Интервал задается минимальным и максимальным целым числом.
- ППЗ-0-8 По данному текстовому файлу выведите в TListView статистику длин строк. А именно в первом столбце – количество символов в строке, во втором – количество раз, сколько раз такая длина встречается. Длина строки не превышает 255 (в противном случае выведите диагностическое сообщение).
- ППЗ-0-9 В текстовом файле в каждой строке указана фамилия учащегося (без пробелов), а затем через пробел оценки учащегося. Выведите в TListView фамилию учащегося, его средний балл и количество оценок (в режиме Report).
- ППЗ-0-10 В текстовом файле уровень вложенности конструкции отражается количеством пробелов в начале строки (как в исходниках программ). Выведите иерархию вложенности строк в TTreeView. Использовать для этого специализированную функцию запрещено.
- ППЗ-0-11 В TListView выведите статистику длин слов, использованных в данном текстовом файле (в строке указывается длина слова, количество раз использования слов такой длины и доля в процентах таких слов).
- ППЗ-0-12 Выведите в TTreeView расписание вашей группы, хранящееся в текстовом файле в удобном вам формате. Без изменения программы это расписание должна быть возможность поменять. В TTreeView в верхнем уровне иерархии указываются дни недели, на второй – номера пар, на третьей – либо предметы, либо информация о четности/нечетности, на последнем уровне – предмет
- ППЗ-0-13 Выведите в TListView таблицу перевода между 10, 2, 8 и 16-ыми числами по введенному пользователю интервалу в 10-ой системе счисления. Интервал задается минимальным и максимальным числом, все переводимые числа – целые. Вывод осуществляется в режиме Report.
- ППЗ-0-14 В файле хранится информация о выборах в следующем формате. В первой строке указывается количество кандидатов  $n$ ; в следующих  $n$  строках – информация о кандидатах (ФИО). В оставшихся строчках файла указывается информация о результатах выборов в округах в формате: номер округа, через пробел результаты каждого участника (порядок их такой же, как и в первых строках файла). Выведите эту информацию в TTreeView: на верхнем уровне иерархии номера участков, на следующем – ФИО кандидатов с их результатами.

- ППЗ-0-15 В файле хранится информация о выборах в следующем формате. В первой строке указывается количество кандидатов  $n$ ; в следующих  $n$  строках – информация о кандидатах (ФИО). В оставшихся строчках файла указывается информация о результатах выборов в округах в формате: номер округа, через пробел результаты каждого участника (порядок их такой же, как и в первых строках файла). Выведите эту информацию в TListView в режиме Report: в первом столбце номер участка, в каждом следующем – результаты участника. Заголовки второго и следующих столбцов должны содержать ФИО кандидатов.
- ППЗ-0-16 **Скобочная структура.** Пользователь в TEdit вводит выражение, содержащее скобки (предполагается, что они открываются и закрываются корректно, но проверять это надо). Выведите в TTreeView это выражение, при этом вложенным скобкам должно соответствовать вложенный узел дерева. В каждый узел выводится содержимое соответствующих скобок. Корневой узел – это все выражение.
- ППЗ-0-17 Выведите в TListView в режиме Report таблицу умножения в системе счисления, выбранной пользователем (от 2 до 35).
- ППЗ-0-18 Выведите в TListView таблицу Брадиса для синусов и косинусов (без ручного или автоматизированного переписывания, конечно): <http://www.vsetabl.ru/056.htm>
- ППЗ-0-19 В текстовом файле хранится информация о расписании электричек на данной станции. Выведите ее в TListView. Формат хранения Вы выбираете сами с учетом того, что он должен быть удобен для редактирования в текстовом редакторе.
- ППЗ-0-20 Выведите в TListView таблицу Брадиса для квадратного корня (без ручного или автоматизированного переписывания, конечно)
- ППЗ-0-21 Выведите в TTreeView все двочные числа от 0 до 1023 следующим способом. Каждое число представляется десятью цифрами (слева добавляются незначащие нули). После этого на верхнем уровне иерархии выбирается первая цифра, на следующем вторая (выводятся обе) и так далее. На 10-м выводятся сами числа и они же в 10-ой системе счисления. Программа должна заполнять TTreeView автоматически (а не показывать заранее введенные значения).
- ППЗ-0-22 Выведите в TListView расписание вашей группы, хранящееся в текстовом файле в удобном вам формате. Без изменения программы это расписание должна быть возможность поменять. В TListView указывается день, номер пары, название предмета/предметов (для нечетной/четной недели) /в режиме Report/.
- ППЗ-0-23 В текстовом файле в каждой строке указаны два комплексных числа в формате  $\text{число}+i\text{число}$  или  $\text{число}-i\text{число}$ . Выведите в TListView в режиме Report исходные числа и их сумму, разность, произведение и отношение.
- ППЗ-0-24 С клавиатуры вводится натуральное число  $n$  (не больше 7). Заполните магический квадрат  $n \times n$  внутри TListView (в режиме Report).
- ППЗ-0-25 Выведите в TTreeView расписание преподавателя, хранящееся в текстовом файле в удобном вам формате. Без изменения программы это расписание должна быть возможность поменять. В TTreeView в верхнем уровне иерархии указываются дни недели, на второй – номера пар, на третьей – предмет и группа либо информация о четности/нечетности, на последнем уровне – предмет и группа
- ППЗ-0-26 В TListView выведите статистику длин строк, использованных в данном текстовом файле (в строке указывается длина строки, количество раз использования строк такой длины и доля в процентах таких строк).
- ППЗ-0-27 В TListView выведите статистику количества слов в строках, заданных в данном текстовом файле (в строке указывается длина строки (в словах), количество раз использования строк такой длины и доля в процентах таких строк).
- ППЗ-0-28 В TListView выведите статистику количества слов в абзацах, заданных в данном текстовом файле (в строке указывается длина абзаца (в словах), количество раз использования абзацев такой длины и доля в процентах таких абзацев). Начало абзаца распознается путем проверки наличия ровно трех пробелов в начале строки.

### Первый уровень

Обратите внимание, что использование каких-либо функций для работы с файлами, кроме стандартных в этом задании запрещено.

- ППЗ-1-1 В текстовом файле записана информация о планируемых мероприятиях в формате:

*дата.месяц.год час:минута название*

На каждой строке описывается одно мероприятие. Дата, месяц, год, час и минута записываются в виде натурального числа.

Выведите в TTreeView информацию о мероприятиях. На верхнем уровне иерархии должны отображаться года, на следующем – месяца, на следующем – даты, на следующем – часы, на следующем – сами мероприятия (с указанием часов и минут)

- ППЗ-1-2 В текстовом файле разделы обозначены следующим образом: каждый раздел обозначается строкой, в начале которой расположено не менее 10 пробелов, а далее указывается его номер в формате *число название* или *число.число название* при этом чисел перед названием может быть сколько угодно (числа все натуральные). Первое число обозначает номер раздела высшего уровня, следующее число - номер подраздела, следующее – номер подраздела подраздела и т. д. Выведите в TTreeView иерархию разделов (и подразделов) данного текстового файла. В случае наличия любой ошибки в файле выведите на экран диагностическое сообщение и прекратите обработку файла
- ППЗ-1-3 Предоставить пользователю путем кликания по рисунку рисовать отрезки, вывести эти отрезки в ListView, предоставить возможность путем выбора отрезка в ListView и нажатия кнопки удалить отрезок. **Задача требует дополнительного изучения обработки событий**
- (а) Выведите с помощью TListView треугольник Паскаля того размера, который выберет пользователь. **Задача требует динамической генерации столбцов**
- ППЗ-1-4 В текстовом файле записана информация о электричках:  
*час:минута пункт отправления | пункт назначения*  
На каждой строке описывается один маршрут электрички. час и минута записываются в виде натурального числа. Выведите в TTreeView информацию об электричках. На верхнем уровне иерархии должны отображаться пункты отправления, на следующем – час, на следующем – минуты и пункт назначения.
- ППЗ-1-5 В каждой строке текстового файла написано слово (внутри него нет пробелов и дефисов) и его определение (после знака -). Выведите в TTreeView на верхнем уровне все первые буквы слов, на втором – первые и вторые и так далее на последнем уровне – сами слова и их определения.
- ППЗ-1-6 В файле записано некоторое количество слов строчными буквами без пробелов и дефисов внутри слов. Выведите эти слова внутри TreeView по следующему правилу: слово находится внутри узла другого слова, если все буквы первого слова есть во втором слове (с учетом числа повторений).
- ППЗ-1-7 В текстовом файле хранятся слова русского языка в формате *буквы(корень)буквы*. Буквы до и/или после корня могут отсутствовать. Выведите в TTreeView все указанные слова, группируя их по корням, а внутри каждой группы по алфавиту. При выводе слов скобки, которые выделяют корни, надо опускать
- ППЗ-1-8 В текстовом файле хранится информация о тарифах сотового оператора в следующем формате. В первой строке указывается количество возможных опций в тарифе *n*. В следующих *n* строках указывается краткое название опции (те же правила, что и у названий переменных Паскаля) и через пробел ее полное наименование (оно ограничивается концом строки). Каждая следующая строка описывает один тариф: до символа | идет название тарифа, а далее через пробелы названия опций и их стоимости. Выведите эту информацию в TTreeView: в верхнем уровне иерархии идет название тарифа, в следующем – название опции и ее стоимость.
- ППЗ-1-9 В текстовом файле хранится информация о некоторых сущностях подобно следующему примеру:  
Тип данных->Скалярный тип->Целочисленный тип->Integer  
Тип данных->Сложный тип->Массив  
Выведите информацию об этих сущностях в TTreeView: в узлах, не имеющих родительских узлов, то, что стоит до первой стрелки, во вложенных узлах – то, что между первой и второй стрелкой, и так далее.
- ППЗ-1-10 В текстовом файле хранится информация в формате CSV, выведите ее в TListView. Запрещено использовать специализированные функции для формата CSV.
- ППЗ-1-11 Из исходного текста программы, написанной на языке Паскаль, выведите содержимое всех комментариев в TListView (при выводе переходы на новые строки игнорируются).
- ППЗ-1-12 Выведите из исходного текста программы на языке C++ содержимое всех комментариев в TListView (при выводе переходы на новые строки игнорируются).

## Второй уровень

Обратите внимание, что использование каких-либо функций для работы с файлами, кроме стандартных в этом задании запрещено.

- ППЗ-2-1 Вывести с помощью TListView информацию о многоугольниках, что описаны в текстовом файле: в каждой строке указываются координаты всех вершин одного многоугольника. В компоненте должно быть три столбца: количество вершин, периметр, площадь. Проверять на то, что многоугольник является корректным многоугольником не требуется.
- ППЗ-2-2 Из текстового файла читаются числа и записываются в AVL-дерево. Отобразите его с помощью TTreeView.
- ППЗ-2-3 В данном текстовом файле найдите повторяющиеся строки и в TListView выведите содержимое этих строк и количество повторений. Проверка будет проводиться на файлах экстремальной длины (больше объема оперативной памяти).
- ППЗ-2-4 Из текстового файла читаются строки и записываются в AVL-дерево. Отобразите его с помощью TTreeView.
- ППЗ-2-5 Без использования встроенных компонент (и специализированных методов) выведите в ListView содержимое текущего каталога (включая названия файлов, их размер, расширение, флаги).
- ППЗ-2-6 Без использования встроенных компонент (и специализированных методов) выведите TreeView содержимое текущего каталога (и его подкаталогов).
- ППЗ-2-7 В каждой строке текстового файла написаны ФИО работника и через знак | ФИО его непосредственного начальника. Выведите структуру предприятия в TTreeView.
- ППЗ-2-8 Выведите по данному каталогу протокол snapshot'ов git в TListView. Выполнение задания означает запуск в данном каталоге команды `git log`, анализ результатов команды и вывод их в TListView (режим Report).

## Третий уровень

Обратите внимание, что использование каких-либо функций для работы с файлами, кроме стандартных в этом задании запрещено.

- ППЗ-3-1 Вывести с помощью TTreeView иерархию процедур и функций программы, написанной на языке Паскаль, без использования ООП (программа выбирается с помощью TOpenDialog).
- ППЗ-3-2 Вывести с помощью TTreeView отобразите синтаксическое дерево по данному корректному выражению языка Паскаль, содержащему операции +, -, \*, /, div, mod и скобки. Операндами могут выступать целые числа или названия переменных. Подсчет при этом проводить не нужно. Само понятие синтаксического дерева можно уточнить у преподавателя
- ППЗ-3-3 Вывести с помощью TTreeView иерархию разделов LaTeX-документа. Обработку include и аналогичных команд проводить не надо
- ППЗ-3-4 Вывести с помощью TTreeView иерархию разделов HTML-документа. Необходимо обрабатывать тэги комментария и тэги H1-H6. Использовать встроенные в язык программирования парсеры (кроме тех, что входят в стандарт языка Паскаль) запрещено.
- ППЗ-3-5 Пользователь указывает основной файл исходного кода программы на языке Паскаль. После чего в TListView для каждого файла исходного кода этой программы указывается какие модули в нем явно используются (подключены с помощью uses). Программа должна анализировать всю иерархию подключенных модулей, исходник которых расположены в том же каталоге, что сама программа.
- ППЗ-3-6 Пользователь указывает основной файл исходного кода программы на языке Паскаль (программа не использует модули). После чего в TTreeView выводится иерархия объектов (- типов данных) этой программы.
- ППЗ-3-7 Выведите информацию из текстового файла в формате JSON в TreeView (с учетом иерархии объектов).
- ППЗ-3-8 Для данного файла сформировать дерево для кода Хаффмана и вывести его в TTreeView. Вы можете выбрать разумное ограничение для размера файла.

### 3. Содержание отчета:

- (a) Исходный текст первой программы
- (b) Ответы на контрольные вопросы

### 4. Контрольные вопросы:

- (a) Опишите методику работы с TTreeView или TListView.

## 6 Практическая работа №4. Динамическое создание элементов управления (4 часа)

1. Цель работы: практическое закрепление знаний о динамическом создании элементов управления (без использования дизайнера)
2. Порядок выполнения:
  - (a) осуществите визуальное проектирование пользовательского интерфейса формы;
  - (b) разработайте методы – обработчики тех событий, что необходимо использовать в вашей программе;
  - (c) приведите вашу программу в соответствие с требованиями.

В задании обязательно необходимо создавать элементы управления «налету».

Варианты заданий:

### 0 уровень

- ПП4-0-1 Пользователь указывает количество складываемых чисел, после чего появляется соответствующее число `TEdit`'ов. После заполнения всех `TEdit`'ов и нажатия кнопки сложить программа должна вычислить сумму введенных чисел. Количество `TEdit`'ов не превышает 20.
- ПП4-0-2 Пользователь указывает количество перемножаемых чисел, после чего появляется соответствующее число `TEdit`'ов. После заполнения всех `TEdit`'ов и нажатия кнопки перемножить программа должна вычислить произведение введенных чисел. Количество `TEdit`'ов не превышает 20.
- ПП4-0-3 Пользователю предлагается ввести для каждого дня предыдущего месяца среднюю температуру. После нажатия кнопки «обработать» ему выводится средняя температура за месяц (элементы управления для ввода температуры и соответствующие `TLabel` создаются в программе, а не рисуются на форме).
- ПП4-0-4 Пользователь указывает разрядность двоичного числа, после чего появляется соответствующее число `TCheckBox`'ов. После выбора некоторых `TCheckBox`'ов и нажатия кнопки «получить число» программа должна вывести полученное число в 10-ой системе счисления. Каждому `TCheckBox`'у соответствует двоичный разряд (включенный `TCheckBox` обозначает цифру 1; выключенный – 0).
- ПП4-0-5 Пользователю предлагается ввести число  $n = 1, \dots, 20$ , после чего появится  $n$  компонентов типа `TEdit` и рядом с ними  $n$  компонентов типа `TCheckBox`. После нажатия кнопки «Вычислить» на форме появляется сумма тех чисел, которые стоят рядом с включенными переключателями.
- ПП4-0-6 Пользователю предлагается ввести число  $n = 1, \dots, 20$ , после чего появится  $n$  компонентов типа `TEdit` и рядом с ними  $n$  компонентов типа `TCheckBox`. После нажатия кнопки «Вычислить» на форме появляется произведение тех чисел, которые стоят рядом с включенными переключателями.
- ПП4-0-7 Пользователь указывает количество чисел, после чего появляется соответствующее число `TEdit`'ов. После заполнения всех `TEdit`'ов и нажатия кнопки «отсортировать» программа должна переставить введенные числа в порядке возрастания. Количество `TEdit`'ов не превышает 20.
- ПП4-0-8 Пользователь выбирает графический файл (формат файла выбирает студент), после чего он появляется на форме. Далее пользователь выбирает, на сколько его частей надо разбить по горизонтали и вертикали, после чего появляется соответствующее число `TImage`'ей, в которых нарисованы соответствующие части картинки (кроме того, эти картинки надо друг от друга отделить).
- ПП4-0-9 Пользователь вводит целое число  $n = 2, \dots, 20$ , после чего появляется соответствующее число `TDateTime`. После нажатия кнопки «вычислить» наибольшее расстояние между введенными временами (в днях, часах, минутах, секундах).
- ПП4-0-10 Пользователь вводит целое число  $n = 2, \dots, 100$ , после чего появляется соответствующее число звездочек (`TImage`), которые падают внутри формы, и после их исчезновения с поверхности формы соответствующие `TImage` уничтожаются.
- ПП4-0-11 Пользователь вводит целое число  $n = 2, \dots, 5$ , после чего появляются соответствующее число компонентов типа `TEdit`, между соседними из них – соответствующее число `TComboBox`, содержащих знаки операций (+, -, \*, /). После ввода чисел и выбора знаков операций пользователь нажимает кнопку «вычислить», в результате чего на форме появляется результат вычисления выражения без учёта приоритета умножения и деления.
- ПП4-0-12 Пользователь вводит количество цифр (от 1 до 7) в 16-ричном числе, после чего появляется соответствующее число компонентов типа `TComboBox`. После нажатия кнопки «перевести» должно появиться введенное число в 2-ой системе счисления.

- ПП4-0-13 Пользователь вводит количество цифр (от 1 до 7) в 8-ричном числе, после чего появляется соответствующее число компонентов типа `TComboBox`. После нажатия кнопки «перевести» должно появиться введенное число в 2-ой системе счисления.
- ПП4-0-14 Пользователь вводит количество цифр (от 1 до 14) в 3-ичном числе, после чего появляется соответствующее число компонентов типа `TComboBox`. После нажатия кнопки «перевести» должно появиться введенное число в 9-ой системе счисления.
- ПП4-0-15 Пользователь вводит количество кругов (от 1 до 14), после чего появляется соответствующее число компонентов типа `TEdit` для ввода абсцисс, ординат и радиусов. После нажатия кнопки «вывести» должны появиться описанные круги (`TShape`).
- ПП4-0-16 Пользователь вводит количество прямоугольников (от 1 до 14), после чего появляется соответствующее число компонентов типа `TEdit` для ввода абсцисс и ординат левого верхнего и правого нижнего угла. После нажатия кнопки «вывести» должны появиться описанные прямоугольники (`TRectangle`).
- ПП4-0-17 Пользователь вводит целое число  $n = 2, \dots, 20$ , после чего появляется соответствующее число `TDateTime`. После нажатия кнопки «вычислить» наименьшее расстояние между введенными временами (в днях, часах, минутах, секундах).

## 1 уровень

- ПП4-1-1 Пользователь указывает размерность прямоугольной таблицы, после чего появляется соответствующее число `TEdit`'ов. После заполнения всех `TEdit`'ов и нажатия кнопки вычислить программа должна вычислить сумму чисел в каждой строке, в каждом столбце, а также сумму всех введенных чисел в целом. Количество `TEdit`'ов не превышает 100 ( $10 \times 10$ ).
- ПП4-1-2 Пользователь указывает размерность прямоугольной таблицы, после чего появляется соответствующее число `TEdit`'ов. После заполнения всех `TEdit`'ов и нажатия кнопки вычислить программа должна вычислить максимальное число среди минимальных элементов строк. Количество `TEdit`'ов не превышает 100 ( $10 \times 10$ ).
- ПП4-1-3 С использованием `TShape` создайте простейший графический редактор, позволяющий разместить фигуру из представленных на панели фигур на рисунке, путем указания координат в `TEdit` и кликанья на рисунок, и удалять фигуры путем двойного клика по фигуре, размещенной на рисунке.
- ПП1-1-4 Программа для изучения ассемблера. Программа должна позволять вводить два двоичных 32-битных числа и выполнять операции `ADD`, `SUB` и показывать флаги: `AF`, `CF`, `SF`, `PF`, `ZF`, `OF`. В этой программе поля для ввода отдельных цифр должны генерироваться программой «налету», а не задаваться в дизайнера.
- ПП4-1-5 В файле, в удобном вам формате, записаны расположение элементов управления `TEdit`. После запуска программы и выбора файла, эти элементы управления появляются на экране; и после каждого изменения чисел, хранящихся в `TEdit`, на форме в `TLabel` появляется сумма введенных чисел.
- ПП4-1-6 В файле, в удобном вам формате, записаны расположение элементов управления `TLabel` и интервалы времени для каждого из них в миллисекундах. После запуска программы и выбора файла, эти элементы управления появляются на экране и с заданными интервалами времени численное значение, хранящееся в `TLabel` увеличивается на один. Здесь предполагается использование равного числа элементов управления типов `TLabel` и `TTimer`.
- ПП4-1-7 В файле хранится текст в следующем формате:

```
Title:Название
Описание
Описание
Описание
Title:Название2
Описание2
Описание2
```

Осуществите выбор файла с помощью `TOpenDialog` и создание `TPageControl` с таким числом страниц, сколько встречается строк, начинающихся на `Title`. В заголовок страницы выведите названия разделов, а на сами страницы – описания.

- ПП4-1-8 Создайте калькулятор для всех систем счисления от 2-ой до 36-ой с возможностью динамически переключаться между ними; при этом кнопки-цифры должны автоматически создаваться/уничтожаться в зависимости от системы счисления. Калькулятор должен позволять проводить сложение, вычитание, умножение, целочисленное деление и нахождение остатка. Вводить отрицательные числа не нужно.

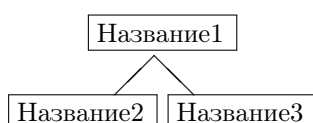
ПП4-1-9 Пользователь вводит целое число  $n = 2, \dots, 5$ , после чего появляются соответствующее число компонентов типа TEdit, между соседними из них – соответствующее число TComboBox, содержащих знаки операций (+, -, \*, /). После ввода комплексных чисел и выбора знаков операций пользователь нажимает кнопку «вычислить», в результате чего на форме появляется результат вычисления выражения без учёта приоритета умножения и деления.

## 2 уровень

ПП4-2-1 В файле задано дерево в формате:

id L:id1 R:id2 Название

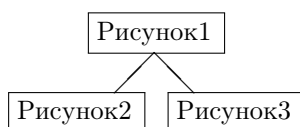
Где id – идентификатор вершины дерева, id1 – идентификатор вершины левого поддерева, id2 – идентификатор вершины правого поддерева, название – текст, выводимый в вершине. Выведите соответствующее дерево в виде TLabel, соединенных линиями, нарисованными на TImage. При этом внешний вид полученного дерева должен быть похож на следующий:



ПП4-2-2 В файле задано дерево в формате:

id L:id1 R:id2 Название файла, хранящего рисунок

Где id – идентификатор вершины дерева, id1 – идентификатор вершины левого поддерева, id2 – идентификатор вершины правого поддерева, файл – рисунок, выводимый в вершине. Выведите соответствующее дерево в виде TImage, соединенных линиями, нарисованными на другом TImage. При этом внешний вид полученного дерева должен быть похож на следующий:



Рисунки можно считать одинаковыми по размеру (или их можно масштабировать). Размер рисунков выбирает студент.

## 3 уровень

ПП4-3-1 Из файла читается формула, после чего создаются TEdit'ы для каждой переменной с указанием в TLabel'e ее названия, а также появляется кнопка вычислить, позволяющая осуществить вычисления по данной формуле с использованием чисел, введенных пользователем.

В формуле могут участвовать операции сложения, вычитания, умножения, деления и скобки; вещественные и целые константы, переменные.

ПП4-3-2 Пользователь указывает размерность вводимой матрицы, после чего появляется соответствующее число TEdit'ов. Далее пользователь вводит элементы матрицы и после нажатия кнопки «вычислить» программа выводит на экран определитель матрицы (можно использовать метод Гаусса или разложение по строкам/столбцам). Размерность матрицы не превышает  $6 \times 6$ .

ПП4-3-3 Пользователь указывает размерность вводимой матрицы, после чего появляется соответствующее число TEdit'ов. Далее пользователь вводит элементы матрицы и после нажатия кнопки «вычислить» программа выводит на экран обратную матрицу. Размерность матрицы не превышает  $6 \times 6$ .

3. Содержание отчета:

(а) Исходный текст первой программы



(b) Ответы на контрольные вопросы

4. Контрольные вопросы:

(a) Опишите методику создания элементов управления без использования дизайнера.